Algebraic Reasoning About Timeliness

by Seyed Hossein HAERI[1,2] Peter W. THOMPSON[1,3] Peter VAN ROY[4] Magne HAVERAAEN[2] Neil J. DAVIES[1,3] Mikhail BARASH[2] Kevin HAMMOND[1] James CHAPMAN[1]
([1]IOG, [2]University of Bergen, [3]PNSol, [4]UCLouvain)

on 19 Jun 2023

16th Interaction and Concurrency Experience
NOVA University, Lisbon, Portugal

Introduction

» Why predict performance?

* Weather forecast of today can't arrive tomorrow!
* Without performance prediction
    * Performance issues exposed late in design cycle
    * Either:
        * Re-architect the design, with cost and delay, or
        * Allocate excessive resources, with cost and inefficiency.
* With performance prediction
    * Performance issues exposed early in design cycle
        * Re-architect the design before time and money spent, and
        * Control resources, avoiding cost and inefficiency.

See [1, §1.1] for more:

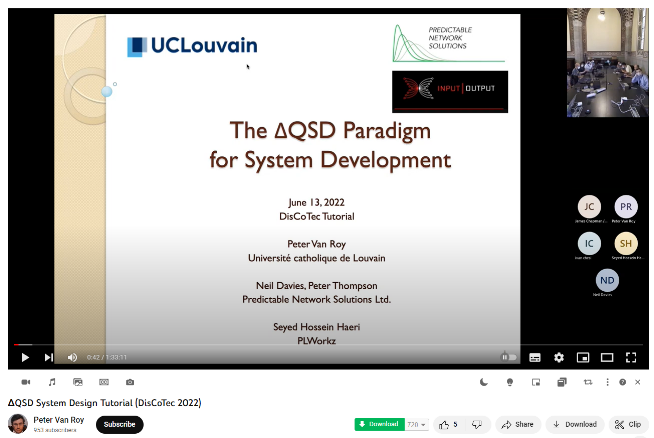
> *Mind Your Outcomes: The ∆QSD Paradigm for Quality-Centric Systems Development and Its Application to a Blockchain Case Study*. Computers 11(3): 45 (2022)
> `https://www.mdpi.com/2073-431X/11/3/45`

» Why does IOG fund research on performance?

- Good Starting Point:
  Kevin Hammond's Keynote in Lambda Days 2023
  `https://tinyurl.com/3t42t3wn`
- IOG is a prominent blockchain company.
  `https://iohk.io`
- The effective operation of the Cardano network depends on a performance aware design.
- PNSol is world-leading performance company.
  `http://www.pnsol.com`
- The ΔQSD Team on Formalising Performance Aspects

» △QSD Advertisement

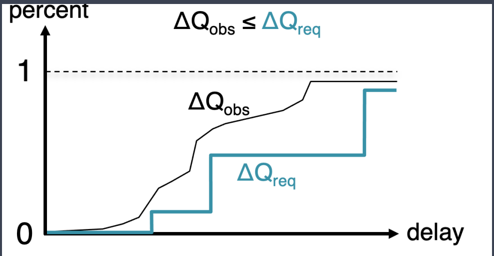
Last Year's DisCoTec Tutorial by Peter VAN ROY



https://www.youtube.com/watch?v=iBYZEJZwKm0

» What's timeliness?

Timeliness
is delivering results within the required time bounds
(sufficiently often).



Cache Example

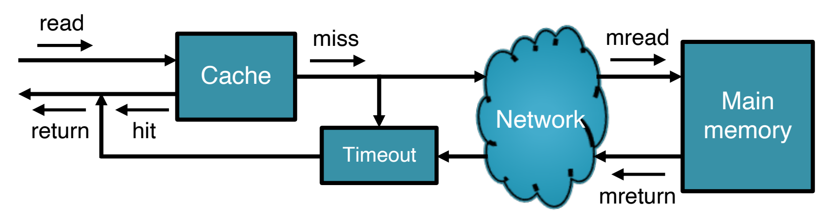
* Outcome Diagrams
* Outcome Expressions
* An Algebraic Perspective on Timeliness
* Where is the algebra?

Cache Example

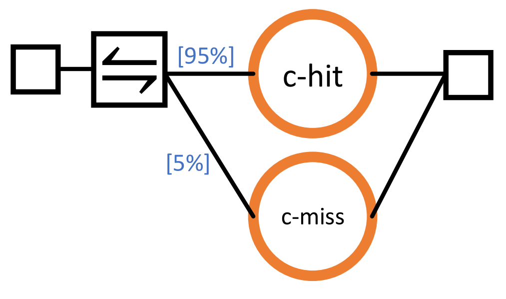
* Outcome Diagrams
* Outcome Expressions
* An Algebraic Perspective on Timeliness
* Where is the algebra?

» Big Picture

Block Diagram



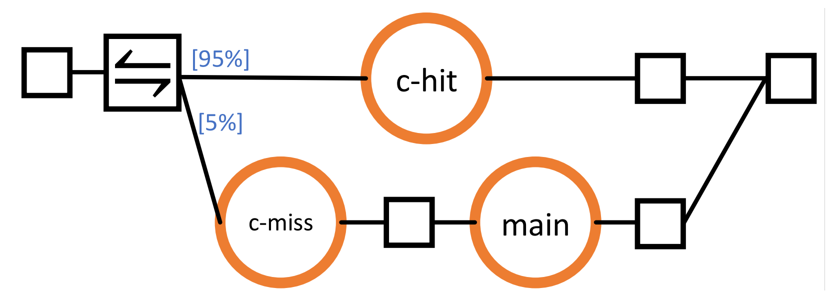
» Hit or Miss



Note:

* Outcomes: What the System Gains by Performing One its Tasks
  * NOT System States
  * NOT Subsystems
  * NOT Classes/Objects
* Probabilistic Choice (⇋)

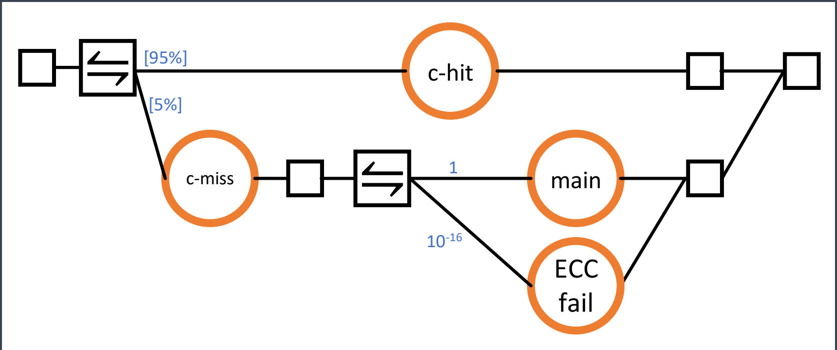
» Lookup from Main Memory



Note:

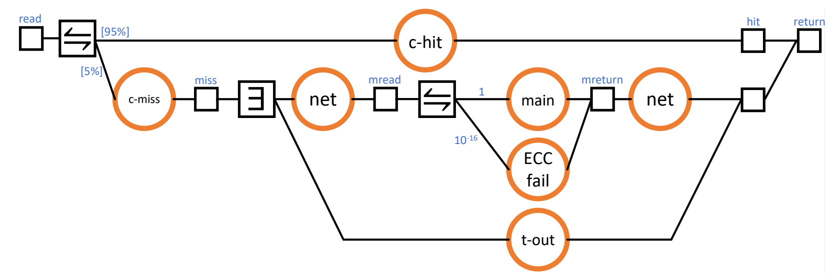
* Sequential Composition
* Left-to-Right Causality

» Error Correction

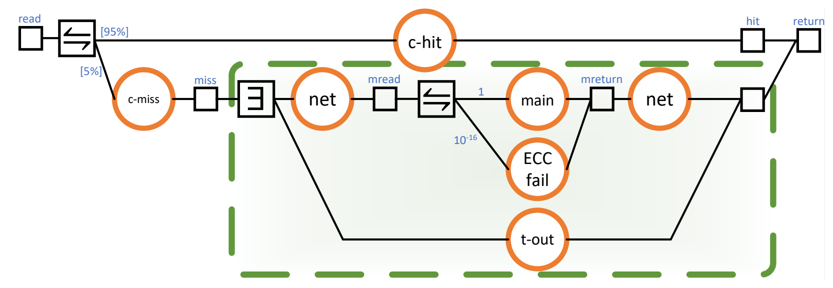


» Timeout (1 of 3)

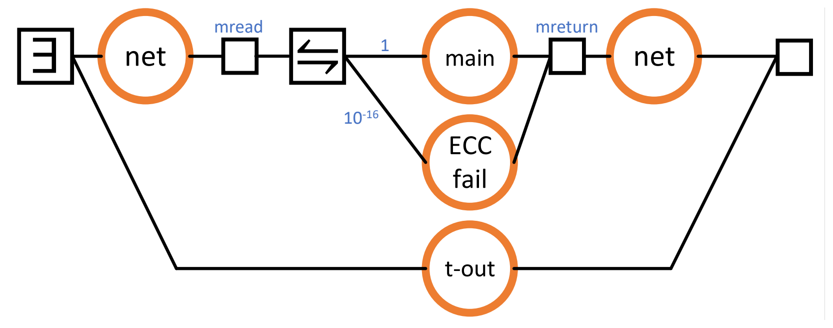
Time-Bounded Network Connection Back & Forth



» Timeout (2 of 3)



» Timeout (3 of 3)



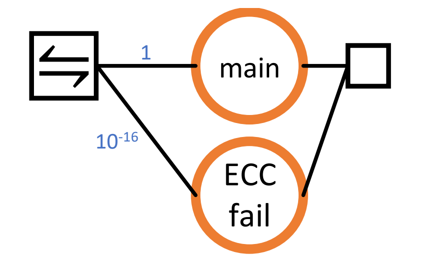
Note:

* Any-to-Finish (∃)

Cache Example

* Outcome Diagrams
* Outcome Expressions
* An Algebraic Perspective on Timeliness
* Where is the algebra?

» ECC Only



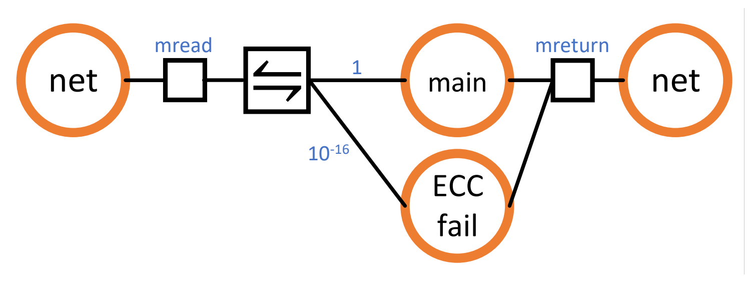
Expression:

$$\text{main} \underset{10^{-16}}{\overset{1}{\leftrightharpoons}} \perp$$

Note:

* "$\perp$" is for failure.

» ECC + Network



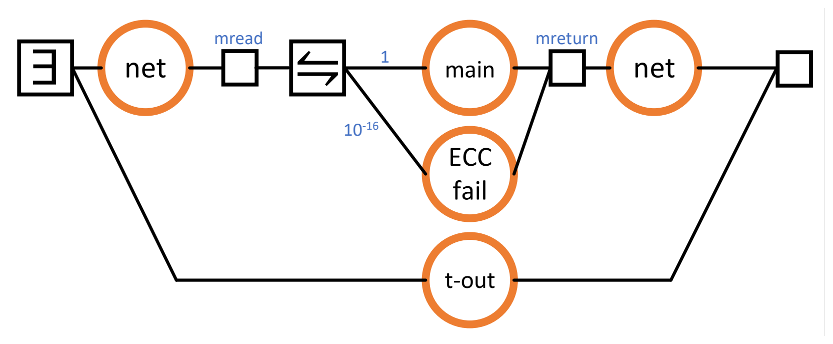
Expression:

$$\text{net} \;\bullet\!\!\rightarrowtail\!\!\bullet\; \left(\text{main} \;\underset{10^{-16}}{\overset{1}{\leftrightharpoons}}\; \perp\right) \;\bullet\!\!\rightarrowtail\!\!\bullet\; \text{net}$$

Note:

* "$\bullet\!\!\rightarrowtail\!\!\bullet$" is for sequential composition.

» ECC + Network + Timeout



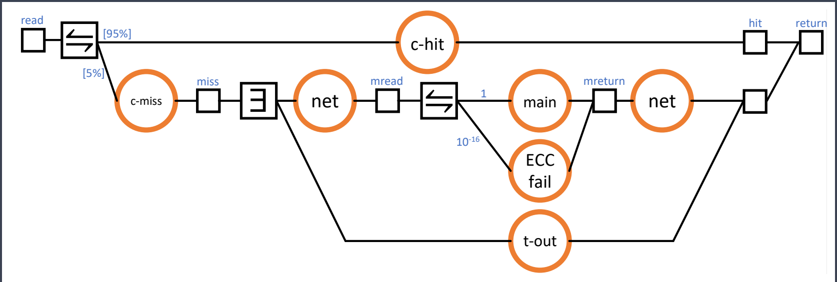
Expression:

$$(\mathsf{net} \bullet\!\!\rightarrow\!\!\bullet (\mathsf{main} \underset{10^{-16}}{\overset{1}{\leftrightarrows}} \bot) \bullet\!\!\rightarrow\!\!\bullet \mathsf{net}) \parallel^{\exists} \mathsf{t\text{-}out}$$

Note:

- “$\parallel^{\exists}$” is for any-to-finish.

» Full Expression



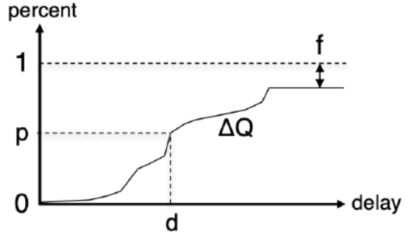
$$\text{c-hit} \overset{[95\%]}{\leftrightharpoons} (\text{c-miss} \bullet\!\!\rightarrow\!\!\bullet ((\text{net} \bullet\!\!\rightarrow\!\!\bullet (\text{main} \underset{10^{-16}}{\overset{1}{\leftrightharpoons}} \bot) \bullet\!\!\rightarrow\!\!\bullet \text{net}) \parallel^{\exists} \text{t-out}))$$

Cache Example

* Outcome Diagrams
* Outcome Expressions
* An Algebraic Perspective on Timeliness
* Where is the algebra?

» What's a △Q?

* Quality Attenuation
* A Measure for Delay (and Failure)
* Represented using a Cumulative Distribution Function (CDF)
* Improper Random Variable (IRV) [2]



» Timeliness Semantics

Definition (Haeri et al. [1]): Given *a* basic assignment $\Delta_\circ[\![.]\!] : \overline{\mathbb{B}} \to \Delta$, define $\Delta\mathtt{Q}[\![.]\!]_{\Delta_\circ} : \mathbb{O} \to \mathbb{I}$ such that

» Timeliness Semantics

Definition (Haeri et al. [1]): Given a basic assignment $\Delta_\circ[\![.]\!] : \overline{\mathbb{B}} \to \Delta$, define $\Delta Q[\![.]\!]_{\Delta_\circ} : \mathbb{O} \to \mathbb{I}$ such that

$$
\begin{aligned}
\Delta Q[\![\beta]\!]_{\Delta_\circ} &= \begin{cases} \mathbf{1} & \text{when } \Delta_\circ[\![\beta]\!] \notin \mathbb{I} \\ \Delta_\circ[\![\beta]\!] & \text{otherwise} \end{cases} \\
\Delta Q[\![o \bullet\!\!\rightarrow\!\!\bullet\, o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} * \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \underset{m'}{\overset{m}{\rightleftharpoons}} o']\!]_{\Delta_\circ} &= \frac{m}{m+m'} \Delta Q[\![o]\!]_{\Delta_\circ} + \frac{m'}{m+m'} \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \parallel^{\forall} o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} \times \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \parallel^{\exists} o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} + \Delta Q[\![o']\!]_{\Delta_\circ} - \Delta Q[\![o]\!]_{\Delta_\circ} \times \Delta Q[\![o']\!]_{\Delta_\circ}
\end{aligned}
$$

» Timeliness Semantics

Definition (Haeri et al. [1]): Given a basic assignment $\Delta_\circ[\![.]\!] : \overline{\mathbb{B}} \to \Delta$, define $\Delta Q[\![.]\!]_{\Delta_\circ} : \mathbb{O} \to \mathbb{I}$ such that

$$
\begin{aligned}
\Delta Q[\![\beta]\!]_{\Delta_\circ} &= \begin{cases} \mathbf{1} & \text{when } \Delta_\circ[\![\beta]\!] \notin \mathbb{I} \\ \Delta_\circ[\![\beta]\!] & \text{otherwise} \end{cases} \\
\Delta Q[\![o \bullet\!\!\rightarrow\!\!\bullet\, o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} * \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \overset{m}{\underset{m'}{\leftrightharpoons}} o']\!]_{\Delta_\circ} &= \frac{m}{m+m'} \Delta Q[\![o]\!]_{\Delta_\circ} + \frac{m'}{m+m'} \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \parallel^{\forall} o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} \times \Delta Q[\![o']\!]_{\Delta_\circ} \\
\Delta Q[\![o \parallel^{\exists} o']\!]_{\Delta_\circ} &= \Delta Q[\![o]\!]_{\Delta_\circ} + \Delta Q[\![o']\!]_{\Delta_\circ} - \Delta Q[\![o]\!]_{\Delta_\circ} \times \Delta Q[\![o']\!]_{\Delta_\circ}
\end{aligned}
$$

» ΔQ of the Cache Example

Given

$$\Delta_\circ \supseteq \{\Delta Q_{\text{c-hit}}, \Delta Q_{\text{c-miss}}, \Delta Q_{\text{mem}}, \Delta Q_{\text{t-out}}, \Delta Q_{\text{mem}}, \Delta Q_{\text{t-out}}\},$$

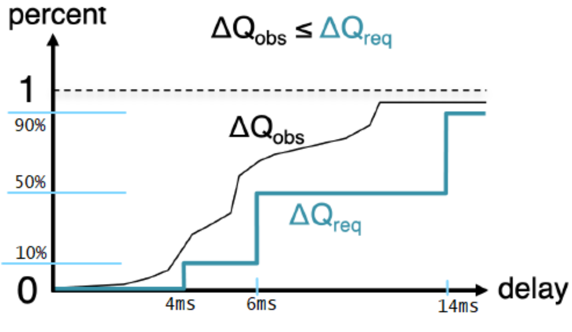
one calculates:

$$\Delta Q_{\text{obs}} = 0.95 \times \Delta Q_{\text{c-hit}} + 0.05 \times (\Delta Q_{\text{c-miss}} * (\Delta Q_{\text{mem}} + \Delta Q_{\text{t-out}} - \Delta Q_{\text{mem}} \times \Delta Q_{\text{t-out}})),$$

where

$$\Delta Q_{\text{mem}} = \Delta Q_{\text{net}} * (1 - 10^{-16}) \times \Delta Q_{\text{main}} * \Delta Q_{\text{net}}.$$

» Timeliness for the Cache



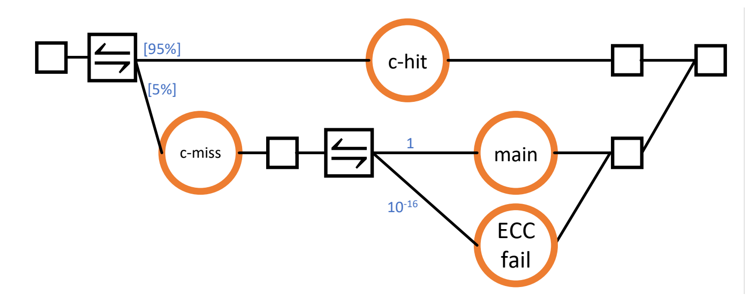
ΔQ_{req}:

- 10% of queries up to 4ms
- 50% of queries up to 6ms
- 90% of queries up to 14ms
- 10% of queries never

Cache Example

* Outcome Diagrams
* Outcome Expressions
* An Algebraic Perspective on Timeliness
* Where is the algebra?

» ECC Revisited



Expression:

$$\text{c-hit} \overset{[95\%]}{\leftrightharpoons} \left(\text{c-miss} \bullet\!\!\rightarrowtail\!\!\bullet \left(\text{main} \underset{10^{-16}}{\overset{1}{\leftrightharpoons}} \bot\right)\right)$$

» Algebraic Manipulation

$$\text{c-hit} \overset{[95\%]}{\leftrightarrows} (\text{c-miss} \bullet\!\!\rightarrow\!\!\bullet (\text{main} \underset{10^{-16}}{\overset{1}{\leftrightarrows}} \bot))$$

$$\text{c-hit} \overset{[95\%]}{\leftrightarrows} ((\text{c-miss} \bullet\!\!\rightarrow\!\!\bullet \text{main}) \underset{10^{-16}}{\overset{1}{\leftrightarrows}} \bot)$$

$$(\text{c-hit} \overset{[\cdot]}{\leftrightarrows} (\text{c-miss} \bullet\!\!\rightarrow\!\!\bullet \text{main})) \overset{[q]}{\leftrightarrows} \bot$$

where $q = (1 - 0.05 \times 10^{-16}) = 0.99999999999999999995.$

- 17 nines vs 9 nines of Ericsson AXD301

Not a Guarantee for Success!

Just ruling out infeasibility with this level of information.

» Benefit of Algebraic Manipulation

$q = (1 - 0.05 \times 10^{-16}) = 0.999999999999999995$

- What if we had already implemented the cache?
- Will simply throwing more hardware at it work?
- Re-architecture from scratch?

Algebraic Results

» Algebraic Structures

$\mathbb{O}$ with	Forms
$\leftrightharpoons$	magma
$\bullet\!\!\rightarrow\!\!\bullet$	**commutative monoid** with $\top$ and $\bot$ as the identity and absorbing elements
$\|^{\forall}$	**commutative monoid** with $\top$ and $\bot$ as the identity and absorbing elements
$\|^{\exists}$	**commutative monoid** with $\bot$ and $\top$ as the identity and absorbing elements

Note:

Neither $\|^{\forall}$ nor $\|^{\exists}$ nor their combination form the familiar richer algebraic structures.

» Equivalences Containing Constant Outcomes

$$\bot \leftrightharpoons \bot = \bot \qquad o \bullet\!\!\rightarrow\!\!\bullet \bot = \bot \qquad \top \leftrightharpoons \top = \top \qquad \bot \bullet\!\!\rightarrow\!\!\bullet o = \bot$$

$$\top \bullet\!\!\rightarrow\!\!\bullet o = o \qquad o \bullet\!\!\rightarrow\!\!\bullet \top = o \qquad \top \|^{\forall} o = o \qquad \bot \|^{\exists} o = o$$

$$o_1 \bullet\!\!\rightarrow\!\!\bullet (o_2 \leftrightharpoons \bot) = (o_1 \bullet\!\!\rightarrow\!\!\bullet o_2) \leftrightharpoons \bot$$

$$(o_1 \leftrightharpoons \bot) \bullet\!\!\rightarrow\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrow\!\!\bullet o_2) \leftrightharpoons \bot$$

$$(o_1 \leftrightharpoons \top) \bullet\!\!\rightarrow\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrow\!\!\bullet o_2) \leftrightharpoons o_2$$

$$o_1 \bullet\!\!\rightarrow\!\!\bullet (o_2 \leftrightharpoons \top) = (o_1 \bullet\!\!\rightarrow\!\!\bullet o_2) \leftrightharpoons o_1$$

$$o_1 \overset{[p]}{\leftrightharpoons} (o_2 \overset{[q]}{\leftrightharpoons} \top) = o_2 \overset{[q(1-p)]}{\leftrightharpoons} (o_1 \overset{\left[\frac{p}{1-q(1-p)}\right]}{\leftrightharpoons} \top)$$

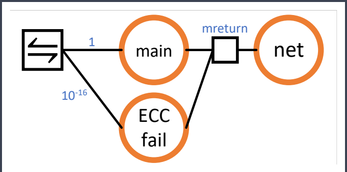
$$\bot \overset{[p]}{\leftrightharpoons} (\bot \overset{[q]}{\leftrightharpoons} o) = \bot \overset{[p+(1-p)q]}{\leftrightharpoons} o$$

» Equivalences Containing Constant Outcomes

$$\bot \leftrightharpoons \bot = \bot \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \bot = \bot \qquad \top \leftrightharpoons \top = \top \qquad \bot \bullet\!\!\rightarrowtail\!\!\bullet o = \bot$$

$$\top \bullet\!\!\rightarrowtail\!\!\bullet o = o \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \top = o \qquad \top \|^{\forall} o = o \qquad \bot \|^{\exists} o = o$$

$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \bot) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$
$$(o_1 \leftrightharpoons \bot) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$
$$(o_1 \leftrightharpoons \top) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_2$$
$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \top) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_1$$
$$o_1 \overset{[p]}{\leftrightharpoons} (o_2 \overset{[q]}{\leftrightharpoons} \top) = o_2 \overset{[q(1-p)]}{\leftrightharpoons} \left(o_1 \overset{\left[\frac{p}{1-q(1-p)}\right]}{\leftrightharpoons} \top\right)$$
$$\bot \overset{[p]}{\leftrightharpoons} (\bot \overset{[q]}{\leftrightharpoons} o) = \bot \overset{[p+(1-p)q]}{\leftrightharpoons} o$$

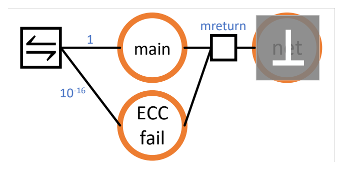


» Equivalences Containing Constant Outcomes

$$\bot \leftrightharpoons \bot = \bot \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \bot = \bot \qquad \top \leftrightharpoons \top = \top \qquad \bot \bullet\!\!\rightarrowtail\!\!\bullet o = \bot$$

$$\top \bullet\!\!\rightarrowtail\!\!\bullet o = o \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \top = o \qquad \top \|^{\forall} o = o \qquad \bot \|^{\exists} o = o$$

$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \bot) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$
$$(o_1 \leftrightharpoons \bot) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$
$$(o_1 \leftrightharpoons \top) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_2$$
$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \top) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_1$$
$$o_1 \overset{[p]}{\leftrightharpoons} (o_2 \overset{[q]}{\leftrightharpoons} \top) = o_2 \overset{[q(1-p)]}{\leftrightharpoons} (o_1 \overset{\left[\frac{p}{1-q(1-p)}\right]}{\leftrightharpoons} \top)$$
$$\bot \overset{[p]}{\leftrightharpoons} (\bot \overset{[q]}{\leftrightharpoons} o) = \bot \overset{[p+(1-p)q]}{\leftrightharpoons} o$$



ECC followed by a net failure is **as timely as** failure itself!

» Equivalences Containing Constant Outcomes

$$\bot \leftrightharpoons \bot = \bot \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \bot = \bot \qquad \top \leftrightharpoons \top = \top \qquad \bot \bullet\!\!\rightarrowtail\!\!\bullet o = \bot$$

$$\top \bullet\!\!\rightarrowtail\!\!\bullet o = o \qquad o \bullet\!\!\rightarrowtail\!\!\bullet \top = o \qquad \top \|^{\forall} o = o \qquad \bot \|^{\exists} o = o$$

$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \bot) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$

$$(o_1 \leftrightharpoons \bot) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons \bot$$

$$(o_1 \leftrightharpoons \top) \bullet\!\!\rightarrowtail\!\!\bullet o_2 = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_2$$

$$o_1 \bullet\!\!\rightarrowtail\!\!\bullet (o_2 \leftrightharpoons \top) = (o_1 \bullet\!\!\rightarrowtail\!\!\bullet o_2) \leftrightharpoons o_1$$

$$o_1 \overset{[p]}{\leftrightharpoons} (o_2 \overset{[q]}{\leftrightharpoons} \top) = o_2 \overset{[q(1-p)]}{\leftrightharpoons} \left(o_1 \overset{\left[\frac{p}{1-q(1-p)}\right]}{\leftrightharpoons} \top\right)$$

$$\bot \overset{[p]}{\leftrightharpoons} (\bot \overset{[q]}{\leftrightharpoons} o) = \bot \overset{[p+(1-p)q]}{\leftrightharpoons} o$$

Seen at the Algebraic Manipulation of the Cache Example

» Distributivity

Theorem

Let $o_1, o_2, o_3 \in$ *and* $p \in \{\bullet\!\!\rightarrow\!\!\bullet, \|^{\forall}, \|^{\exists}\}$*. Then,*

* $\odot\odot$ *time* $\models o_1 \; p \; (o_2 \leftrightharpoons o_3) = (o_1 \; p \; o_2) \leftrightharpoons (o_1 \; p \; o_3)$, *and*
* $\odot\odot$ *time* $\models (o_1 \leftrightharpoons o_2) \; p \; o_3 = (o_1 \; p \; o_3) \leftrightharpoons (o_2 \; p \; o_3)$.

Bad News! Only 3 Out of the Possible 15

» Summary

- Formalisation of ΔQSD – Ongoing Project
- Algebraic Manipulations $\Rightarrow$ Tool Support
- Properisation
  - Ordinary $\Delta Q[\![.]\!]$ doesn't work!
  - The First IRV Body of Theorems Ever!

Q&A

» Questions?

» Thank you very much!

S. H. Haeri, P. Thompson, N. Davies, P. Van Roy, K. Hammond, and J. Chapman.
Mind Your Outcomes: The ΔQSD Paradigm for Quality-Centric Systems Development and Its Application to a Blockchain Case Study.
Computers, 11(3):45, 2022.

K. S. Trivedi.
Probability and Statistics with Reliability, Queuing, and Computer Science Applications.
Wiley, New York, NY, USA, 2 edition, 2002.